

Your Ref: 07844-468JP1  
Our Ref: PA994

**Translation of Selected Portions of  
Pat. Laid-open Official Gazette**

-----  
Appln. No: 6-304326  
Appln. Date: November 14, 1994  
Laid-open Pub. No: 8-137871  
Laid-open Pub. Date: May 31, 1996

Inventor(s): Hiroshi Satomi  
Applicant(s): Canon K.K.  
Attorney(s): Toshihiko Watabe  
-----

**1. Title of the Invention**

**DOCUMENT CREATING APPARATUS**

**2. Claims**

(omitted)

**3. Detailed Description of the Invention (Selected Portions)**

1)

(omitted)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-137871

(43) 公開日 平成8年(1996)5月31日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 17/25

識別記号

庁内整理番号

F I

技術表示箇所

9288-5L

G 0 6 F 15/20

5 4 2 A

審査請求 未請求 請求項の数1 F D (全13頁)

(21) 出願番号

特願平6-304328

(22) 出願日

平成6年(1994)11月14日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 里見 宏

東京都大田区下丸子3丁目30番2号 キヤ

ノン株式会社内

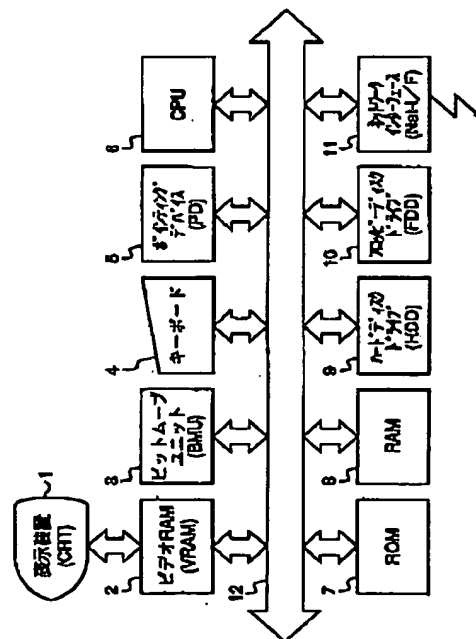
(74) 代理人 弁理士 渡部 敏彦

(54) 【発明の名称】 文書作成装置

(57) 【要約】

【目的】 文書編集者の意図通りに文字を配置できる文書作成装置を提供する。

【構成】 領域内に配置された文字データの一部に文字間隔を変更する修飾情報をCPU6が設定し、文字データを配置する時に前記修飾情報が設定されているか否かをCPU6が判定し、前記修飾情報が設定されている文字データを配置するためにCPU6が文字間隔を変更し、CPU6が文字データを行内に配置し、該配置された文字データを表示装置1が表示する。



## 【特許請求の範囲】

【請求項1】 領域内に配置された文字データの一部に文字間隔を変更する修飾情報を設定する修飾情報設定手段と、前記文字データを配置する時に前記文字間隔を変更する修飾情報が設定されているか否かを判定する判定手段と、前記文字間隔を変更する修飾情報が設定されている文字データを配置するために前記文字間隔を変更する文字間隔変更手段と、前記文字データを行内に配置する文字データ配置手段と、該文字データ配置手段によって配置された文字データを表示する表示手段とを具備したことを特徴とする文書作成装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、文字と文字の間隔を自由に変更し、表示し得るようにした文書作成装置に関する。

## 【0002】

【従来の技術】 従来のこの種の文書作成装置における文字配置決定方法は、次のようになっている。即ち、まず、行頭の文字配置位置を決定する。次に文字を順次配置する。この時、文字の配置位置は、直前の文字の配置位置に直前の文字の文字幅を加算した位置に確定される。文字フォントによっては、フォント自体に文字と文字の間隔を付加している場合もあるが、このような場合には、直前の文字の配置位置に直前の文字の文字幅と現在の文字のフォントに設定されている字間を加算した位置に確定される。

【0003】 また、文書作成装置によっては、各文字と文字の間に一定の字間を定義できるものもあるが、その場合も、直前の文字の配置位置に直前の文字の文字幅と一定の字間を加算した位置に確定される。

【0004】 また、一部の文書作成装置では、行末で追い出しや追い込みがあった場合には、行末を揃えるために行内最後の文字の後ろと行末の座標の差分を、各文字の間に均等に割り振ることによって、行内の均等割り付け処理を行っているものもある。

【0005】 更に、字間を設定できる文書編集装置として、文章中の一部の文字列を指定して、その文字列の文字間隔を狭めるような指定を行うことができる機能を持った文書作成装置は存在した。

## 【0006】

【発明が解決しようとする課題】 しかしながら、上述した従来装置にあっては、文字配置位置は文書作成装置が自動調整を行うことによって完全に固定してしまうため、文章入力者が意図した位置に文字を配置しようとしても、文章入力者の意図通りには配置できないという問題点があった。特に、文字間隔を狭める方向には、文字配置を自由に変更することができたが、文字間隔を広げる方向には自由に設定することはできなかった。

【0007】 そのため、図17の(A)、(B)に示す

ように文書データ中の一部の領域だけ、字間を開けて文字を配置しようとしたり、図18の(A)、(B)に示すように文書の見出しのみに幅を一定に保ったレポートを書こうとした時には、文章編集者が以下のような編集作業を行っていた。

(1) 各文字と文字の間に空白文字を挿入することによって、文字間隔を大きくする(図19の(A)から(B)にする方法)。

(2) 文字間隔が開き過ぎる場合には、各文字と文字の間に挿入する空白文字のフォントを変更して、小さな空白文字を挿入したことにして文字間隔を調整する(図19の(A)から(C)にする方法)。

(3) 文字を配置したい場所にタブ位置を設定し、各文字と文字の間にタブを挿入することによって、文字間隔を一定に保つ(図19の(A)から(D)にする方法)。

【0008】 しかし、上記図19の(A)から(B)、(A)から(C)及び(A)から(D)の各対処による文字間隔の調整を行った場合には、以下のような問題点があった。

(イ) 図20は、文字と文字の間に空白文字を挿入して文字間隔を調整する場合の問題点を表す図である。文書作成装置で使えるフォントの種類には限度があるために、空白文字で任意の幅を作ることができず、微調整ができない。即ち、図20の(A)は、空白のサイズが少し足りない場合を示し、図20の(B)は、配置したい文字位置を示し、図20の(C)は、1ポイント大きくすると長すぎる場合を示している。

(ロ) 図21も、文字と文字の間に空白文字を挿入して文字間隔を調整する場合の問題点を表す図である。文字間隔が広い時には、多数の空白文字を挿入しなければならず、編集作業が面倒である。

(ハ) 図23は、文字と文字の間に空白文字を挿入して文字間隔を調整する場合の編集作業に関する問題点を表す図である。図22の(A)の文章の先頭行に編集作業を行う文字列が増えた場合、図22の(B)に示すように字間の調整に使われた空白文字が改行されて行頭にきてしまったために、段落字下げされたように見えてしまう。この場合には、その空白文字は、図22の(C)に示すように編集者が削除して文字配置位置を調整しなければならない。

(ニ) 図23は、タブ位置を決定することによって、文字間隔を調整する場合の編集作業に関する問題点を表す図である。図23の(A)の文章の先頭行に編集作業を行う文字列が増えた場合、図23の(B)に示すように字間の調整に使われた増減のあった文字数分だけ移動してやらないと、文字配置位置は狂ってしまう。

【0009】 本発明は上述した従来の技術の有するこのような問題点に鑑みてなされたものであり、その目的とするところは、一般の文字に対する修飾情報と同様に簡

単な手順で設定する修飾情報として文字間隔を設定することによって、局所的に文字の配置位置を変更することができる文書作成装置を提供しようとするものである。

【0010】

【課題を解決するための手段】上記目的を達成するために本発明の文書作成装置は、領域内に配置された文字データの一部に文字間隔を変更する修飾情報を設定する修飾情報設定手段と、前記文字データを配置する時に前記文字間隔を変更する修飾情報が設定されているか否かを判定する判定手段と、前記文字間隔を変更する修飾情報が設定されている文字データを配置するために前記文字間隔を変更する文字間隔変更手段と、前記文字データを行内に配置する文字データ配置手段と、該文字データ配置手段によって配置された文字データを表示する表示手段とを具備したことを特徴とするものである。

【0011】

【作用】領域内に配置された文字データの一部に文字間隔を変更する修飾情報を修飾情報設定手段が設定し、前記文字データを配置する時に前記文字間隔を変更する修飾情報が設定されているか否かを判定手段が判定し、前記文字間隔を変更する修飾情報が設定されている文字データを配置するために文字間隔変更手段が前記文字間隔を変更し、文字データ配置手段が前記文字データを行内に配置し、該文字データ配置手段によって配置された文字データを表示手段が表示する。

【0012】

【実施例】以下、本発明の実施例を図面に基づき説明する。

【0013】（第1実施例）まず、本発明の第1実施例を図1乃至図15に基づき説明する。図1は、本発明の第1実施例に係る文書作成装置の構成を示すブロック図であり、同図中、1は表示装置（CRT）で、編集時の文書、各種のメッセージメニュー等を画面に表示する。2はビデオRAM（VRAM）で、表示装置1の画面に表示されるイメージデータを展開記憶する。3はビットマップユニット（BMU）で、メモリ間或はメモリと各デバイス間のデータ転送をコントロールする。4はキーボードで、文書編集等を行うための各種のキーを備えている。5はポインティングデバイス（PD）で、表示装置1の画面上のアイコン等を指し示すために用いられる。

【0014】6はCPU（中央演算処理装置）で、制御プログラムに基づいて本装置の各部を制御する。7はROM（読み出し専用メモリ）で、CPU6が実行する各種のプログラム（文書編集に係るプログラムやエラー処理プログラムをはじめ、後述する本発明の文書作成装置による文書処理を実現するためのプログラム等）が記憶されている。8はRAM（読み書き可能メモリ）で、上述した各種のプログラムをCPU6が実行する時に、ワークエリア、エラー処理時の一時退避エリアとして用い

られる。9はハードディスク（HD）を駆動するハードディスクドライブ（HDD）、10はフロッピーディスク（FD）を駆動するフロッピーディスクドライブ（FDD）である。前記ハードディスク及びフロッピーディスクは、後述するアプリケーションプログラムやデータ、ライブラリー等の保存用に用いられる。11はネットワークインターフェース（Net-I/F）で、他の文書処理装置との間でネットワークを経由してデータ転送を行うために、該ネットワーク上のデータ制御及び診断等を行う。12は上述した各ユニット間を接続するI/O（入出力）バスで、アドレスバス、データバス及び制御バスからなる。

【0015】尚、ROM7に記憶されているプログラムは、本装置に直接接続されているハードディスクやフロッピーディスク等の記憶媒体に記憶されていてもよい。また、前記プログラムは、ハードディスクやフロッピーディスク等の記憶媒体やネットワークを介してシステムや装置に供給することができる。

【0016】＜文字の入力編集処理＞本実施例の文書作成装置における文字の入力編集処理は、従来のワードプロセッサ等で実現されている方法と同様で、ある領域内に従来の組版技術、例えば、改行や自然改行による行送り等により、文字を配置することによって実現する。

【0017】＜字間変更修飾指定処理＞図1の表示装置1の画面に表示されている文字列に対して、図1のキーボード4からのキー入力、または、図1のポインティングデバイス5を使用することによって、修飾を行うべき文字列を範囲指定し、その文字列に対して字間変更処理を実行する。その具体的状態を図4に示す。図4の

（A）は、範囲指定前の状態を、図4の（B）は、範囲指定状態を、図4の（C）は、文字間隔変更後（字間変更処理後）の状態をそれぞれ示す。変更されるべき字間は、図5に示す字間変更パネルから入力される。変更された情報は、文章データの中に埋め込まれて、図1のRAM8上に保存される。その時の文章データのフォーマットについては、ここでは触れない。

【0018】＜文字配置に必要な行の情報＞図2は、本実施例の文書作成装置において文字を配置するために必要な行に関する情報を示す図である。図2の2aは行に設定される種々の情報を示す行状態フラグ（line\_flag）である。行内に配置される文字は、図2の2bで示される行開始座標（kumisitan）から図2の2cで示される行末座標（kumisyuuta）までの間に置かれることになる。図2の2cで示される行内文字数（word\_count）は、行内に配置された文字の文字数を保存する領域である。図2の2dで示される行内文字数（word\_count）は、行内に配置された文字の文字数を保存する領域である。図2の2eで示される文字列の長さ（length）は、行内に文字を配置している過程での、配置された文

字列の長さの和を格納する作業領域である。図2の2fで示される行のベースラインは、配置する行全体の基本座標(baseline)を示す領域である。

【0019】＜文字配置に必要な文字の情報＞図3は、本実施例の文書作成装置において文字を配置するために必要な文字に関する情報を示す図である。各々の文字が、以下の情報を保持している。

【0020】図3の3aで示される文字サイズ(word\_size)は、各文字データの大きさを格納する。図3の3bで示される字間(gap)は、行内に文字を配置する時に、配置処理中の文字とその直前の文字との間隔を格納する。図3の3cで示される文字コード(code)は、配置する文字のコードである。図3の3dで示される文字オフセット(offset)は、配置処理中の文字の文章データ中での位置を表している。図3の3eで示される文字配置位置(position)は、行内での文字の配置及び表示位置を表している。図3の3fで示される文字状態フラグ(word\_flag)は、各文字に設定されている修飾情報の状態を示すものである。

【0021】＜行内組版処理＞次に、行内組版処理の概要について、図6を用いて説明する。図6は、行内組版処理の制御手順を示すフローチャートである。

【0022】まず、ステップS601で行の初期化処理を行う。ここでは、「baseline」の設定を行う。その具体的な状態を図12に示す。

【0023】次にステップS602で行内文字配置処理を行って、1行内に配置できる文字を決定する。この処理の詳細は、図7を用いて後述する。

【0024】次にステップS603で行末禁則処理を行う。この処理では、行から溢れている文字の削除処理や禁則処理を行う。本発明においては、この処理は重要ではないので、その詳細については、ここでは触れない。

【0025】次にステップS604で行内の文字の均等割り付け処理を行う。この処理の詳細は、図9を用いて後述する。

【0026】次にステップS605で文字の配置位置の決定処理を行った後、本処理動作を終了する。この文字配置位置決定処理の詳細は、図10を用いて後述する。

length<kumisyuutan-kumisitan... (1)

この判定は、図1のRAM8上の情報を用いて図1のCPU6で実行される。

【0036】上記(1)式が成立せず且つ処理中の文字コードが改行コードであれば、行内に文字が配置できなくなったものとみなして、本処理動作を終了する。また、上記(1)式が成立し且つ処理中の文字コードが改行コードでなければ、まだ行内に文字が配置できるものとみなしてステップS707の処理を行う。このステップS707では、「offset」を「1」加算した後、前記ステップS702へ戻る。この処理は、図1の

【0027】以上が行内組版処理の概要である。

【0028】次に本実施例の文書作成装置による行内文字配置処理について、図7を用いて説明する。

【0029】まず、ステップS701で行に文字を配置する前の初期化処理を行う。初期化するべき情報は、図2に示す行情報である。ここで、「kumisitan」、「kumisyuutan」を設定する。その値に関しては、設定される行の状況によるので、ここでは触れない。また、「word\_count」及び「length」を「0」で初期化する。更に、先頭文字の「offset」を設定する。これらの処理は図1のCPU6で実行され、その結果は、図1のRAM8上に保存される。この時の具体的な状態を図12に示す。

【0030】次にステップS702で、これから配置するべき文字の文字情報を格納する領域を確保する。この領域は、図3に示す情報を格納する領域のことであり、図1のRAM8上に確保される。

【0031】次にステップS703で、前記ステップS702において確保された領域に文字情報を設定する。この処理の詳細は、図8を用いて後述する。設定された情報は、図1のRAM8上に確保される。この時の状態を具体的に表した図が図13である。

【0032】次にステップS704で、行内に配置される文字列の長さ(length)を設定する。「length」は、その文字を配置する前までに配置されていた文字列の長さに、現在配置した文字のサイズ(word\_size)と字間(gap)を加算した長さである。この処理は、図1のCPU6で実行され、その結果は、図1のRAM8上に保存される。この時の状態を具体的に表した図が図13である。

【0033】次にステップS705で、行内文字数(word\_count)を「1」加算する。

【0034】次にステップS706で、現在配置処理中の文字の次の文字を行内に配置できる状態にあるか否かを判定する。この判定は、下記(1)式が成立するか否か、または、処理中の文字コードが改行コードであるか否かである。

【0035】

CPU6で実行され、加算された値は、図1のRAM8上に再度保存される。

【0037】以上の処理により、行内に配置される文字を確定することができる。

【0038】次に、文字情報の設定処理について図8を用いて説明する。図8は、文字情報の設定処理の制御手順を示すフローチャートである。

【0039】図8のステップS801～ステップS803は、従来の文書作成装置における文字の配置処理で使われる手順と同じである。

【0040】まず、ステップS801で、これから処理を行う文字の「offset」を設定する。この処理で設定された値は、図1のRAM8上に再度保存される。

【0041】次に、ステップS802で、文章データから「offset」の位置にある文字を取得し、その文字コード（code）を設定する。この処理で設定された値は、図1のRAM8上に再度保存される。

【0042】次に、ステップS803で、前記ステップS802において設定された「code」を基に、その文字を表示するのに必要な文字サイズ（word\_size）及び前の文字との距離（gap）を取得する。この処理は、図1のCPU6で実行され、設定された値は、図1のRAM8上に再度保存される。

【0043】次に、ステップS804で、処理中の文字に字間変更コマンド（字間変更する旨の修飾情報）が設定されているか否かを判定する。そして、字間変更コマンドが設定されていないと判定された場合には、本処理動作を終了する。また、字間変更コマンドが設定されていると判定された場合には、ステップS805で、現在処理中の文字が先頭の文字であるか否かを判定する。行の先頭文字であった場合には、字間は設定しない方が文字の配置処理としては標準的である。そのため、現在処理中の文字が先頭文字であった場合には、本処理動作を終了する。

【0044】一方、前記ステップS805において現在処理中の文字が先頭の文字でないと判定された場合には、ステップS806で直前の文字も字間変更中か否かを

$$\text{distance} = (\text{kumisyuutan} - \text{kumisitan}) - \text{length} \dots (2)$$

次にステップS902で、行内で均等割り付け処理できる文字数（equality\_count）を求める。この処理の詳細は、図10を用いて後述する。

【0049】次にステップS903で、均等割り付け処理を行うか否か、即ち、equality\_count=0か否かを判定する。そして、equality\_count=0であれば、均等割り付け処理する文字がないので、本処理動作を終了する。

【0050】また、equality\_count=0でなければ均等割り付け処理を行う。そのため、ステップS904で、均等割り付けできる文字に割り付ける長さ（word\_pitch）を求める。「word\_pitch」は、上記（2）式で求めた「distance」を均等割り付けする文字数（equality\_count）で割った値である。

【0051】次にステップS905で、行の先頭の文字を取得する。また、処理中の文字を数える「count」を「0」で初期化する。

【0052】次にステップS906で、その文字の字間を変更できるか否かを判定する。字間変更されている文字については、前記図8のステップS807において

を判定する。そして、直前の文字が字間変更中でない（字間変更の設定がない）と判定された場合には、何もせずに本処理動作を終了する。これは、字間変更の設定が有効なのは、範囲指定された文字列内にある間隔だけであるため、字間変更のかかっている最初の文字は字間変更されないためである。また、前記ステップS805において、直前の文字が字間変更中であると判定された場合には、ステップS807で「gap」を変更すべき字間で置き換え、更に「gap」の置き換えを行ったことを示すために「word\_flag」に「1」をセットした後、本処理動作を終了する。

【0045】以上の処理により、文字情報を設定することができる。

【0046】次に行内文字の均等割り付け処理について、図9を用いて説明する。この処理を具体的に示す図が、図14である。図14の（A）は割り付ける前の状態を、図14の（B）は割り付け後の状態をそれぞれ示す。図9は、行内文字の均等割り付け処理の制御手順を示すフローチャートである。

【0047】まず、ステップS901で、行内文字の均等割り付け処理を行うべき領域の長さ（distance）を算出する。この長さ（distance）は、行頭位置（kumisitan）と行末位置（kumisyuutan）、行内に配置された文字列の長さ（length）から、下記（2）式により求められる。

【0048】

「ward\_flag」に「1」が設定されていた。ここでは、そのフラグが立っているか否かによって判定を行う。ward\_flag=1であれば字間の変更はできない。また、ward\_flag=0であれば字間の変更はできる。

【0053】字間が変更できる文字に関しては、ステップS907で、「gap」に「word\_pitch」を加算した後、ステップS908の処理を行う。また、字間が変更できない文字に関しては、ステップS907をスキップしてステップS908の処理を行う。

【0054】ステップS908では、行内に次の文字があるか否かを判定する。この判定は、下記（3）式が成立するか否かである。

$$\text{count} < \text{ward\_count} \dots (3)$$

上記（3）式が成立しない場合には、行内に文字はないとみなして、本処理動作を終了する。また、上記（3）式が成立する場合には、ステップS909で次の文字を取得して、「count」を「1」進めた後、前記ステップS906へ戻る。

【0056】以上の処理により、行内文字の均等割り付

けが行える。

【0057】次に行内均等割り付け処理できる文字数の取得処理について、図10を用いて説明する。図10は、行内均等割り付けできる文字数の取得処理の制御手順を示すフローチャートである。

【0058】まず、ステップS1001で、この処理で使う作業変数count（処理中の文字数）及びequality\_count（均等割り付けできる文字数）をそれぞれ初期化する。

【0059】次にステップS1002で、行内の先頭の文字情報を取得する。

【0060】次にステップS1003で、現在処理中の文字が均等割り付けできる文字か否かを判定する。この判定は、現在処理中の文字が行頭の文字か否かと、文字情報の「word\_flag」が「1」であるか否かである。行頭の文字は、通常字間を設定しないので均等割り付けできる文字とはみなさない。また、字間変更指定がなされている文字には、「word\_flag」に「1」が設定されている。この文字も均等割り付けできないとみなす。

【0061】均等割り付けできる文字の場合にはステップS1004で、「equality\_count」を「1」増やした後、ステップS1005の処理を行う。また、均等割り付けできない文字の場合には、前記ステップS1004をスキップしてステップS1005の処理を行う。

【0062】ステップS1005では、まだ行内に文字があるか否か、即ち、count<word\_countか否かを判定する。そして、count<word\_countが成立しない時には、行内の文字を全て判定し終ったものとみなして、本処理動作を終了する。また、まだ行内に文字がある場合には、count<word\_countが成立しているため、その時はステップS1006で、次の文字を取得し、「count」を「1」進めた後、前記ステップS1003へ戻る。

【0063】以上の処理によって、行内で均等割り付けできる文字数を取得することができる。

【0064】次に行内の文字配置位置決定処理について、図11を用いて説明する。

【0065】文字の配置位置は、直前の文字の配置位置に直前の文字の文字幅を加算した位置に確定される（フォントによっては、フォント自体に文字と文字の間隔を付加している場合もあるが、このような場合には、直前の文字の配置位置に直前の文字の文字幅と現在の文字のフォントに設定されている字間を加算した位置に確定される）。この処理を具体的に示した図が図15である。

【0066】文字の表示は、ここで求められた配置位置に基づいて表示される。文字の表示位置を表す作業変数として、「work」を使用する。

【0067】図11は、行内の文字配置位置決定処理の

制御手順を示すフローチャートである。

【0068】まず、ステップS1101で、作業変数を初期化する。最初の文字の表示位置は、「kumisitani」であるので、work=kumisitaniである。また、処理中の文字数を数えるための作業変数（count）を「0」で初期化する。

【0069】次にステップS1102で、行内の先頭の文字を取得する。

【0070】次にステップS1103で、現在処理中の文字の表示位置は「work」に格納されているので、この値を文字情報の「position」にセットする。また、次の表示位置を求めるために、「work」に「word\_size」を加算する。

【0071】次にステップS1104で、行内に次の文字があるか否かを判定する。そして、次の文字がなければ本処理動作を終了する。また、次の文字があればステップS1105で、次の文字の文字情報を取得する。

【0072】次にステップS1106で、前記ステップS1105において取得した文字の表示位置を求めるために、「work」に「gap」を加算し、「count」を「1」進めた後、前記ステップS1103へ戻る。

【0073】以上の処理によって、文字の表示位置を決定することができる。

【0074】上述した各種の処理を行うことによって、文字間隔の変更処理は全て終了し、文章入力者の意図する文字間隔を設定した文字配置を行うことができる。

【0075】（第2実施例）上記第1実施例では、行末の均等配置処理で字間変更の設定されている文字の字間は固定したままで、残りの文字の均等配置処理を行った。この場合、1行を越える文字に字間変更を設定した場合等に行末が揃わなくなることがある。行末を揃えて表示したい場合には、上記第1実施例のような設定では、かえって文章の美しさを損なってしまう。

【0076】そこで、第2実施例では、行末揃えを優先する場合と、設定字間を優先する場合を、行の状態フラグによって置き換えることによって、行末を揃えて表示したい場合に、文章の美しさを損なうことがないようにしたものである。

【0077】以下、本発明の第2実施例について、図16を用いて説明する。

【0078】＜文字の入力編集処理＞本実施例における文字の入力編集処理は、上記第1実施例と同一であるから、その説明を省略する。

【0079】＜字間変更修飾指定処理＞本実施例における字間変更修飾指定処理は、上記第1実施例と同一であるから、その説明を省略する。

【0080】＜文字配置に必要な行の情報＞本実施例における文字配置に必要な行の情報において、上記第1実施例と異なる点は、図2の2aで示す行に設定される種

々の情報を示す行状態フラグ (line\_flag) である。本実施例においては、line\_flag=1の時には、行末揃えを優先し、また、line\_flag=0の時には、字間設定による字間を優先する。その他の行情報は、上記第1実施例と同一であるから、その説明を省略する。

【0081】＜文字配置に必要な文字の情報＞本実施例における文字配置に必要な文字の情報は、上記第1実施例と同一であるから、その説明を省略する。

【0082】＜行内文字配置処理＞本実施例における行内組版処理の概要、行内文字配置処理及び文字情報の設定処理は、上記第1実施例における行内組版処理の概要（図6）、行内文字配置処理（図7）及び文字情報の設定処理（図8）と同一であるから、その説明を省略する。

【0083】次に本実施例における1行内での均等割り付け処理について、図9及び図16を用いて説明する。

【0084】図9において、本実施例と第1実施例とで異なる点は、ステップS902及びステップS906の2箇所だけである。

【0085】本実施例では、ステップS902で均等割り付けできる文字数を取得する。

【0086】本実施例における1行内での均等割り付けできる文字数の取得処理について、図16を用いて説明する。図16は、本実施例の文書作成装置における均等割り付けできる文字数の取得処理の制御手順を示すフローチャートである。

【0087】まず、ステップS1601で、この処理で使う作業変数count（処理中の文字数）及びequality\_count（均等割り付けできる文字数）をそれぞれ初期化する。

【0088】次にステップS1602で、行内の先頭の文字情報を取得する。

【0089】次にステップS1603で、現在処理中の文字が行頭の文字か否かを判定する。行頭の文字でない場合には、ステップS1604の処理に移り、また、行頭の文字の場合には、ステップS1607の処理に移る。

【0090】ステップS1604では、line\_flag=1であるか否かを判定する。そして、line\_flag=0の場合には、ステップS1605の処理に移り、また、line\_flag=1の場合には、字間変更中の文字でも行末均等配置の影響を受けさせるためにステップS1606の処理に移る。

【0091】ステップS1605では、word\_flag=1であるか否かを判定する。「word\_flag」は、字間変更の設定がなされている文字のみ「1」がセットされている。この判定を行う場合が、本実施例の特徴である。

【0092】ステップS1606では、均等割り付けで

きる文字数 (equality\_count) を「1」増やし、「word\_flag」に2のビットを立てた後、ステップS1607の処理に移る。

【0093】ステップS1607では、行内にまだ文字があるか否か、即ち、count<word\_countか否かを判定する。count<word\_countが成立しない場合には、行内の文字を全て判定し終ったものとみなして、本処理動作を終了する。行内にまだ文字がある場合には、count<word\_countが成立しているので、その時はステップS1608で次の文字情報を取得した後、前記ステップS1603へ戻る。

【0094】以上の処理によって、行内で均等割り付けできる文字数を取得することができる。

【0095】再び図9の説明に戻る。

【0096】ステップS906で現在処理中の文字が均等割り付けできる文字であるか否かを判定するが、この判定を図16のステップS1606の処理に合わせる。つまり、「word\_flag&2」のビットが立っているか否かで判定する。そして、「word\_flag&2」=2であれば図9のステップS907の処理に移り、均等割り付けを行うための字間を設定する。また、「word\_flag&2」=0であれば図9のステップS908の処理に移る。

【0097】以上の処理によって、字間変更を行いながら、行末揃えを行うように文字配置の調整を行うことができる。

【0098】

【発明の効果】以上詳述したように、本発明の文書作成装置によれば、文書編集者は、文字と文字との間に空白文字を入れたり、その空白文字のサイズを変更したり、また、前後の文章の編集によって文字配置が崩れた場合には、再度その領域の調整をするというような労力を払うことなく、局所的に文字の配置位置を自由に調整することができ、これにより、文書編集者の意図通りに文字を配置することができるという効果を奏する。

【図面の簡単な説明】

【図1】本発明の第1実施例に係る文書作成装置の構成を示すブロック図である。

【図2】図1の文書作成装置における文字の配置に必要な行の情報を示す図である。

【図3】図1の文書作成装置における文字の配置に必要な文字の情報を示す図である。

【図4】図1の文書作成装置における字間変更コマンドの入力処理状態を示す図である。

【図5】図1の文書作成装置における字間入力パネルを示す図である。

【図6】図1の文書作成装置における行内組版処理の制御手順を示すフローチャートである。

【図7】図1の文書作成装置における行内文字配置処理



の制御手順を示すフローチャートである。

【図8】図1の文書作成装置における文字情報の設定処理の制御手順を示すフローチャートである。

【図9】図1の文書作成装置における行内均等割り付け処理の制御手順を示すフローチャートである。

【図10】図1の文書作成装置における均等割り付けできる文字数の取得処理の制御手順を示すフローチャートである。

【図11】図1の文書作成装置における行内の文字配置位置決定処理の制御手順を示すフローチャートである。

【図12】図1の文書作成装置における行情報の初期化状態を具体的に示す図である。

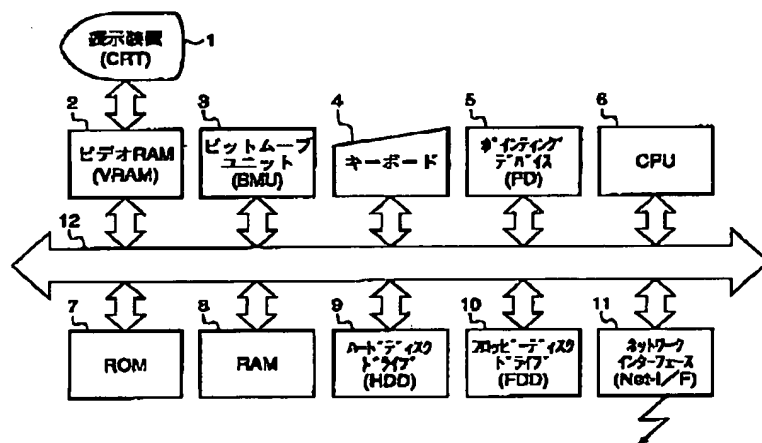
【図13】図1の文書作成装置における文字の配置と文字情報を具体的に示す図である。

【図14】図1の文書作成装置における行の均等配置処理を具体的に示す図である。

【図15】図1の文書作成装置における文字の配置位置決定処理を具体的に示す図である。

【図16】本発明の第2実施例に係る文書作成装置における均等割り付けできる文字数の取得処理の制御手順を

【図1】



【図3】

文字サイズ	word_size	3a
字間	gap	3b
文字コード	code	3c
文字オフセット	offset	3d
文字配置位置	position	3e
文字状態フラグ	word_flag	3f

示すフローチャートである。

【図17】従来の文書作成装置による文字間隔の設定処理を示す図である。

【図18】従来の文書作成装置による文字間隔の設定処理を示す図である。

【図19】従来の文書作成装置による文字間隔の設定処理の問題点を説明するための図である。

【図20】従来の文書作成装置による文字間隔の設定処理の問題点を説明するための図である。

【図21】従来の文書作成装置による文字間隔の設定処理の問題点を説明するための図である。

【図22】従来の文書作成装置による文字間隔の設定処理の問題点を説明するための図である。

【図23】従来の文書作成装置による文字間隔の設定処理の問題点を説明するための図である。

【符号の説明】

1 表示装置（表示手段）

6 CPU（制御情報設定手段、判定手段、文字間隔変更手段、文字データ配置手段）

【図2】

行状態フラグ	line_flag	2a
行開始座標	kumishitan	2b
行末座標	kumisyuntan	2c
行内文字数	word_count	2d
文字列の長さ	length	2e
行のベースライン位置	baseline	2f

【図5】

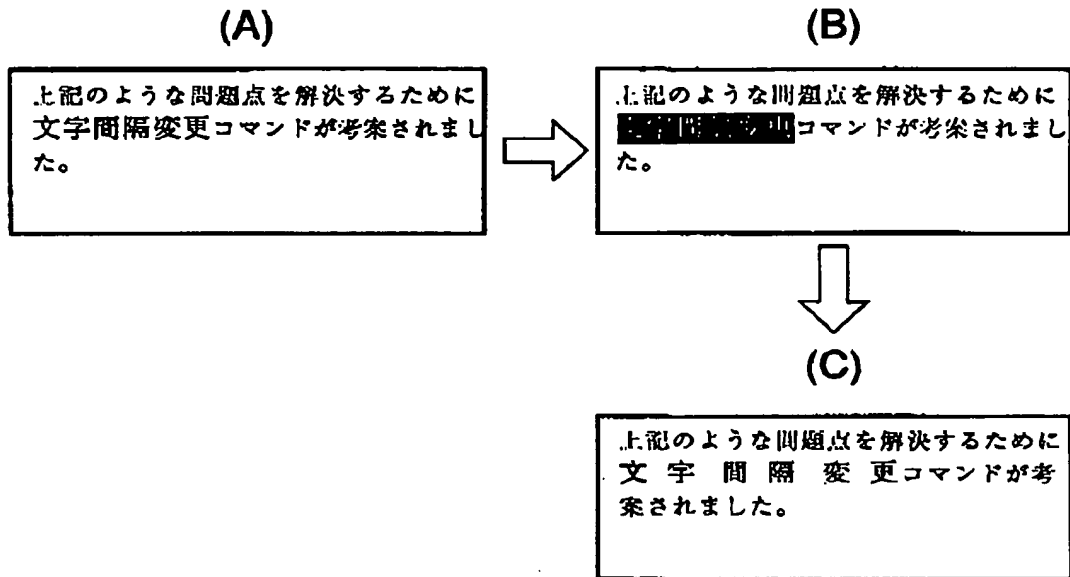
文字サイズ	word_size	3a
字間	gap	3b
文字コード	code	3c
文字オフセット	offset	3d
文字配置位置	position	3e
文字状態フラグ	word_flag	3f

**文字間隔の変更**

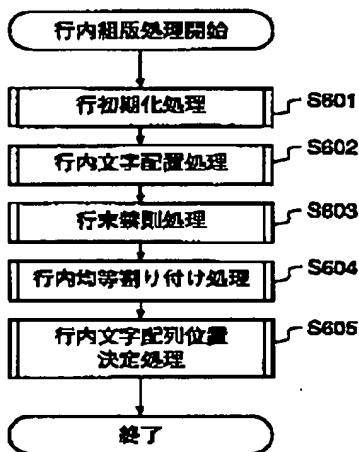
文字の間隔 : 単位

0.5    mm

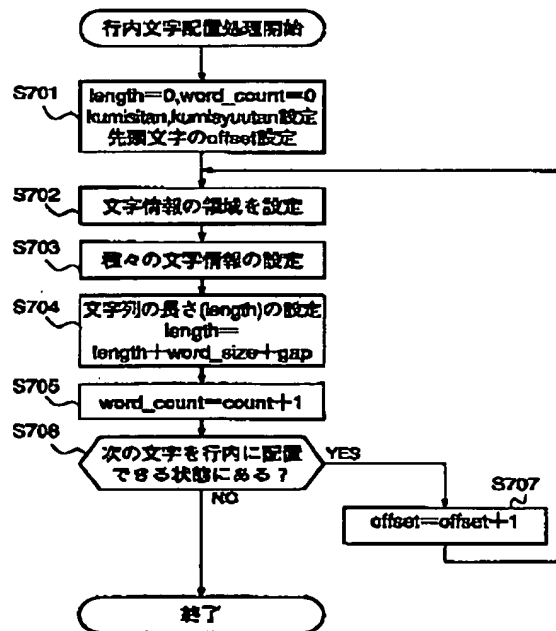
【図4】



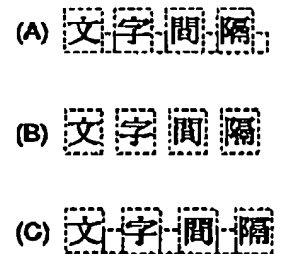
【図6】



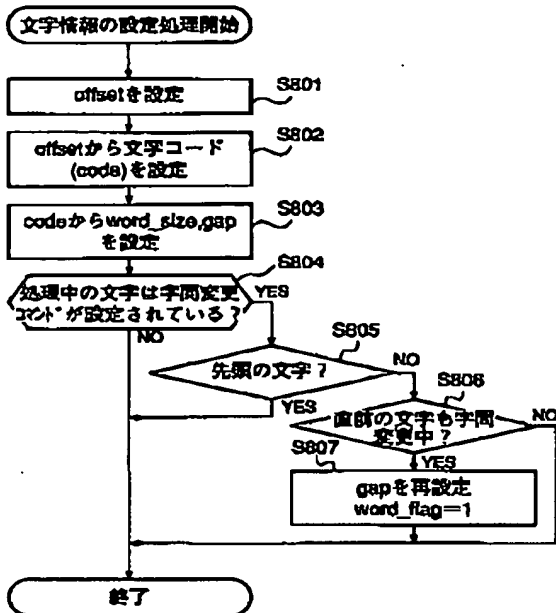
【図7】



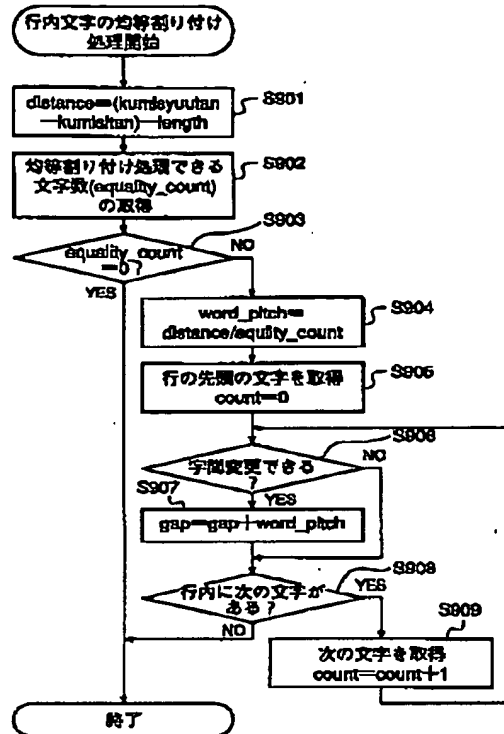
【図20】



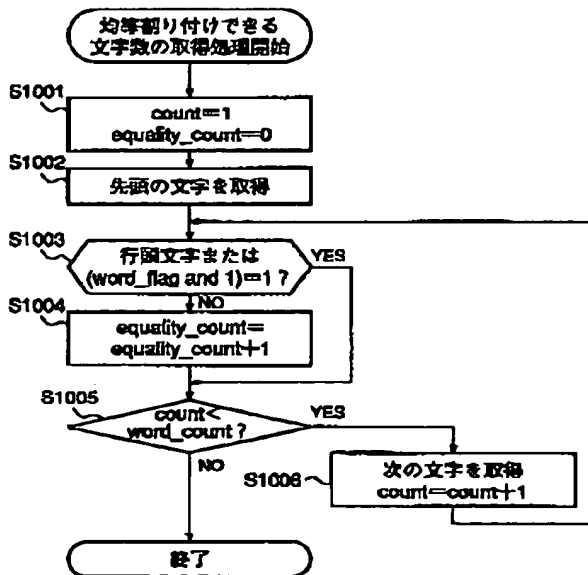
【図 8】



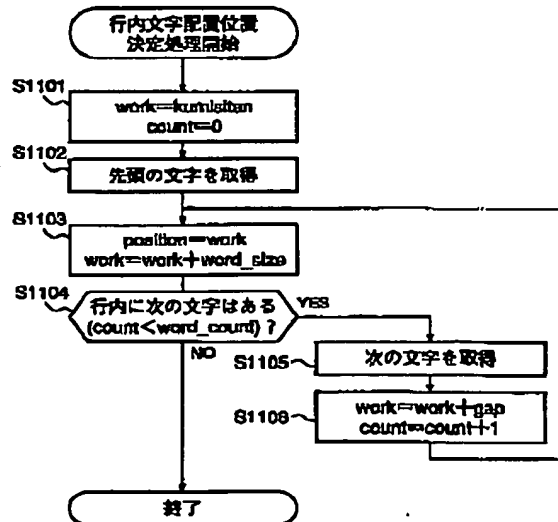
【図 9】



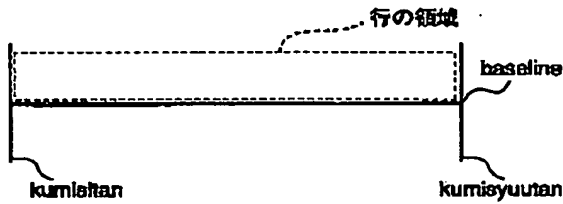
【図 10】



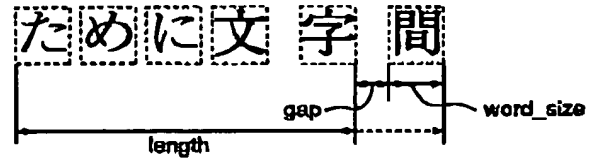
【図 11】



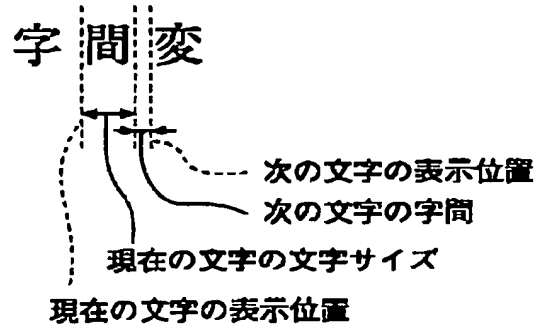
【図12】



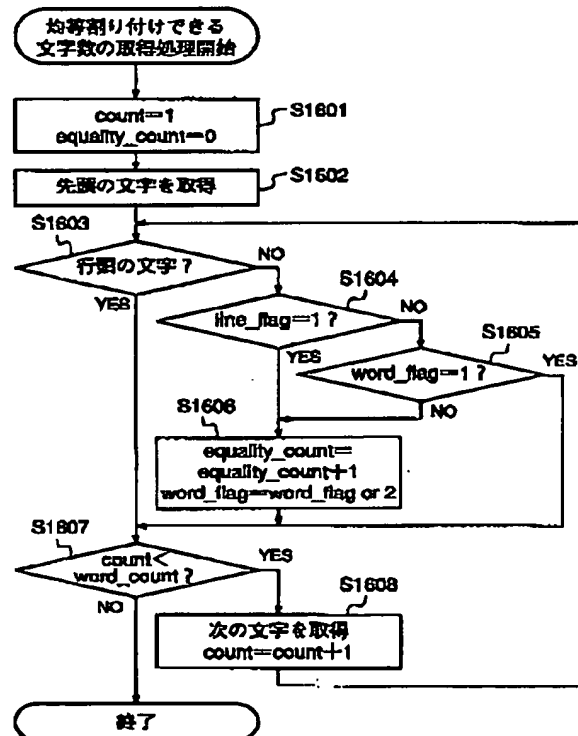
【図13】



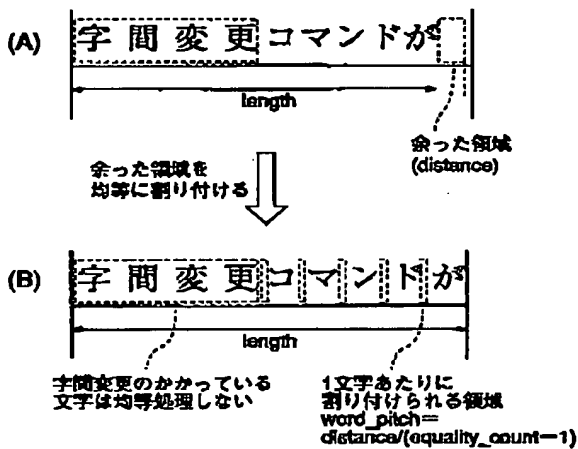
【図15】



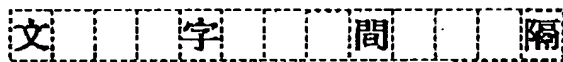
【図16】



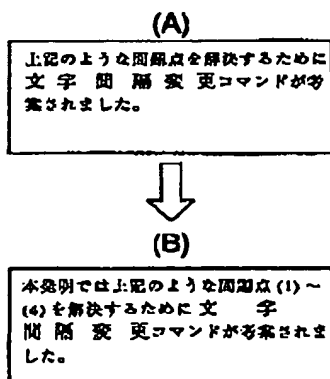
【図14】



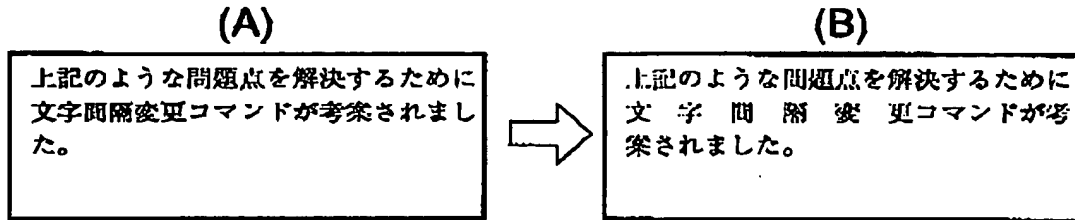
【図21】



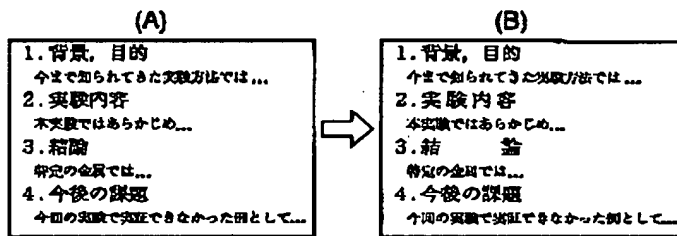
【図23】



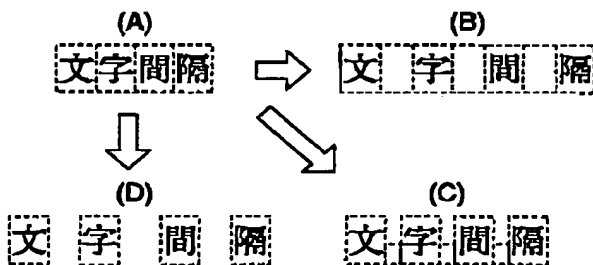
【図17】



【図18】



【図19】



【図22】

